OIPE IAP58
OCT 16 2006
PATENT & TRADEMARK OFFICE

Doc Code: AP.PRE.REQ

PTO/SB/33 (07-05)
Approved for use through xx/xx/200x. OMB 0651-00xx
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

| PRE-APPEAL BRIEF REQUEST FOR REVIEW | Docket Number (Optional) 1801270.00122US1 |
|---|---|

| | Application Number | Filed |
|---|---|---|
| | 09/827,970-Conf. #5419 | April 6, 2001 |
| | First Named Inventor: Alasdair Rawsthorne et al. | |
| | Art Unit: 2128 | Examiner: T. Q. Phan |

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
Note: No more than five (5) pages may be provided.

I am the

- ☐ applicant /inventor.
- ☐ assignee of record of the entire interest. See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)
- ☐ attorney or agent of record. Registration number _____
- ☒ attorney or agent acting under 37 CFR 1.34. Registration number if acting under 37 CFR 1.34. 42,478

Signature

Ronald R. Demsher
Typed or printed name

(617) 526-6000
Telephone number

October 11, 2006
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

☐ *Total of 1 forms are submitted.

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the U.S. Postal Service on the date shown below with sufficient postage as First Class Mail, in an envelope addressed to: MS AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Dated: October 11, 2006 Signature: _____ (Lisa A. Calder)

Claims 1-15 are pending in this application.

SUMMARY OF PROSECUTION HISTORY

On December 22, 2004, the Examiner issued a non-final Office Action rejecting claims 1 and 14 under 35 U.S.C. §112, and claims 1-15 under 35 U.S.C. §103(a) as unpatentable over Kelly (US Patent 6,199,152) in view of Souloglou (Publications 2003/0159134, 2004/0205733, 2004/0210880). In a Response filed on May 26, 2005, the Applicant amended claims 1, 3-5, 10, 11 and 14, and presented arguments that the Examiner found persuasive.

The Examiner issued another non-final Office Action on August 25, 2005, objecting to FIG. 4 and rejecting claims 1-15 under 35 U.S.C. §103(a) as unpatentable over Kelly (US Patent 6,199,152) in view of Le (6,631,514). In a Response filed on January 24, 2006, the Applicant argued against the rejections with respect to Kelly and Le, and submitted a replacement drawing for FIG. 4.

On April 11, 2006, the Examiner issued a final Office Action, repeating the rejections from the Office Action issued on August 25, 2005. The Examiner stated that the replacement drawing for FIG. 4 introduces new matter. In an after-final Response filed on July 11, 2006, the Applicant asked the Examiner to address the arguments presented in the Applicant's response of January 24, 2006. The Applicant also presented reasons why the objection to FIG. 4 should be withdrawn

On August 10, 2006, the Examiner issued an Advisory Action, which addressed the rejections re: Kelly and Le, and withdrew the objection to FIG. 4. The Examiner's explanation on the continuation sheet was somewhat confusing to the Applicant.

On September 13, 2006, an Examiners Interview was conducted at the USPTO including Examiner Thai Phan, Ronald Demsher, Ian Robinson and Alasdair Rawsthorne.

THE CURRENT STATE OF PROSECUTION

Claims 1-15 stand rejected under 35 U.S.C. §103(a) as unpatentable over Kelly (US Patent 6,199,152) in view of Le (6,631,514). The Examiner properly points out that Kelly does not expressly disclose mapping registers in an alternative manner as claimed (see page 3, lines 4-5, of Office Action mailed April 11, 2006). To supply that which is missing from Kelly, the Examiner turns to Le. As is described in more detail below, the Applicant asserts that Kelly cannot be combined with Le in the manner set forth by the Examiner. Further, a combination of Kelly with Le does not teach or suggest all of the limitations set forth in the claims.

THE KELLY PATENT (6,199,152). Kelly discloses an emulator having a fixed, permanent mapping to a set of "working" registers and a set of "official" registers. These roles are fixed and do not ever change.

Kelly provides a detailed introduction to the field of emulators as a type of computer software to enable programs written in the instruction set architecture of a particular type of computer to run on a different type of computer (e.g. emulating a RISC type processor using a CISC type processor or vice versa); see Kelly, columns 1-9.

Kelly describes an emulator, which combines specially adapted "morph host" hardware with emulating "code morphing" software. That is, the emulator considered by Kelly requires the special hardware of the morph host as an essential element of the described emulator.

In Kelly, the "target" processor is the processor for which the target instructions were originally created. The "code morphing software" of Kelly translates these target instructions into host instructions, which are then executed by the "morph host" hardware. In the emulator of Kelly, the morph host hardware provides a large number of hardware registers, including "a set of host or working registers for processing the host instructions and a set of target registers to hold the official state of the target processor for which the target application was created" (column 12, lines 53-57).

That is, in Kelly a target register required for emulation is duplicated in hardware as part of the morph host. Kelly represents each target register by a pair of hardware registers. One of the pair is permanently designated as the "official" target register and the other of the pair is permanently designated as the "working" register. At appropriate points during translation, i.e., when a particular section of host instructions has executed successfully, the content of every "working" register is copied into the corresponding "official" register using the specially adapted copying hardware, in the form of a dedicated interface that "allows an operation called "commit" to quickly transfer the content of all working registers to official target registers and allows an operation called "rollback" to quickly transfer the content of all official target registers back to their working register equivalents," see column 12, lines 57-64. This feature is further explained particularly at col. 16, line 41 to col. 17, line 13. The passages the Examiner cites simply reinforce this understanding of the teachings of Kelly: see col. 15, lines 26-46; col. 16, lines 20-40; col. 23, lines 1-11; and col. 32, lines 40-64.

Kelly clearly teaches that the "working" registers are always the "working" version of the registers, and that the "official" registers always contain the official version of the registers. That is, the "working" and "official" roles of these hardware registers in Kelly are fixed and do not ever change.

There is no motivation for the person skilled in the art to adapt or change the hardware-oriented code morphing system of Kelly. In fact, Kelly teaches that using the dedicated interface in the specially adapted morph host hardware is essential for transferring the contents of the "working" registers into the corresponding "official" registers and that this is a complete and successful solution in itself.

Kelly is equivalent in disclosure to US 5,832,205 cited as background art on page 5, lines 13-29 of the present application. In making the present invention, the inventors identified that the copying operation described in Kelly to copy from the "working" registers to the "official" registers is in fact a significant overhead in the emulation process and could desirably be improved. Secondly, the special hardware features including the "dedicated interface" and additional hardware registers are required by the specially adapted morph host hardware of Kelly. These special hardware features are not available in standard microprocessors, and the inventors realized that it would be desirable to instead provide an emulator which works efficiently even on the standard non-modified hardware of a commonly available processor (e.g., a standard x86 processor).

THE LE PATENT (6,631,514). Le discusses "register renaming" as a form of code optimization, but this register renaming is irrelevant to the use of "official" and "working" registers in an emulator, as recited in the claims.

The present claims recite "alternating mapping," i.e., swapping or reversing the mapping, so that the "official" ("definitive" as used in the claims) and "working" ("speculative" as used in the claims) roles are swapped between first and second locations. As the Examiner points out, this step of "alternating mapping" is not taught or suggested by Kelly. However, Le does not supply that which is missing from Kelly.

The independent claims 1-15 are directed to a method of representing a register in an emulator in which an abstract register representing a subject register of a subject machine (i.e., the "target" register of Kelly) is mapped to either a first location or to a second location within a target ("host") machine. Also, in the present invention, the method comprises alternating mapping of the abstract register between the first and second locations. The content of one of the first or second locations represents a definitive ("official") version of the abstract register for use by the emulator during exception handling, while the other of the first or second locations represents a speculative ("working") version of the abstract register. However, the step of alternating mapping means that the two locations swap roles. The location that was the definitive "official" register is swapped to become the speculative "working" version, and vice versa. The Applicant's claimed method is advantageous in that a definitive version of the emulated register is always available for exception handling, while avoiding the expensive and time-consuming copying operation performed by the specially adapted morph host hardware of Kelly. The Applicant's method reduces the overhead of the emulator and, further, enables the emulator to run more efficiently on standard non-modified hardware (i.e. a standard processor).

Le discloses an emulator that dynamically translates instruction code written for a first architecture, into code for a second architecture. The emulator designates various checkpoints in the original code and speculatively re-orders the translated code instructions according to optimization procedures. If a trap (i.e., an exception) occurs during execution of the re-ordered code, then the emulator resets the original code to the most recent checkpoint. As an optimization to this process, Le teaches register re-naming and storing register maps (RMAPs) associated with particular instructions.

As a particular example, Le discloses that a register Rx in executable object code for a legacy instruction set architecture is mapped using a register map (RMAP) to a corresponding native register rx. There is a one-to-one initial mapping from the legacy register Rx to its corresponding physical architectural register rx. Figure 2C of Le shows that in order to take advantage of instruction level parallelism (ILP) in the native system, the emulator/translator uses register re-naming to break all write after read (WAR) and write after write (WAW) register dependencies, allowing loads and most other operations to be scheduled speculatively. These temporary results are stored in additional temporary registers, on the assumption that the native system has a greater number of registers than the legacy architecture.

Every time a register is renamed, the current RMAP entry is saved and then associated with a particular instruction. If an exception (a trap) occurs then the translator reverts back to the most recent

checkpoint for this section of code, using the register state defined by the checkpoint's RMAP. The translator then proceeds as if the optimization had never taken place. The portions of Le cited by the Examiner simply confirm the analysis presented above. See col. 6 lines 1-5; col. 6 lines 4-8 & lines 60-66; col. 8 line 57 to col. 9 line 15; and col. 9 line 65 to col. 10 line 7.

Notably, Le essentially employs register re-naming. By updating the register map (RMAP), there is always an exact one-to-one relationship between a register Rx of the legacy architecture and exactly one valid register in the native target architecture. Register renaming is a totally separate concept and is not linked at all to the need for "official" and "working" hardware registers for exception handling as in Kelly.

More importantly, Le does not teach or suggest that a representation of a subject register (i.e., an abstract register) from a subject machine is mapped to "either a first location or a second location" within a target machine, as recited in the present claims. Further, Le does not teach or suggest alternating mapping such that the roles of the first and second locations are reversed or swapped, with one of those first or second locations holding the definitive "official" version and the other holding a speculative "working" version. The Examiner points to portions of Le (e.g., col. 6 lines 4-8 & lines 60-66; col. 8 line 57 to col. 9 line 15; and col. 9 line 65 to col. 10 line 7) to teach "alternating mapping of the abstract register between a first location and a second location" as recited in the claims. However, there are two problems with the Examiner's argument. First, the text cited in Le does not teach or suggest alternating of any sort of mapping, as required by the claims. Second, the text cited in Le does not teach or suggest alternating between locations, one of which represents a definitive version and the other represents a speculative version, as required by the claims.Since the alternating mapping to a first location or a second location, recited in the independent claims 1, 9, 12, 13, 14 or 15 is not taught or suggested by Kelly or Le, alone or in combination, those claims should be allowable.

In summary, there is no suggestion or motivation in the Kelly or Le citations themselves or generally in the art to modify the cited references or combine their teachings. The register renaming optimization of Le is wholly unlike the "dedicated interface" of the specially adapted morph host hardware of Kelly. Further, combining the teachings of Kelly with the teachings of Le has no reasonable expectation of success. Finally, as demonstrated above, combination of Kelly with Le does not teach or suggest all of the recited features of the claims.

The dependent claims each recite further new and non-obvious features of the invention over the teachings of the cited Kelly and Le references. For example, claim 4 recites that the alternating mapping step is performed "only if the content of the speculative version of the abstract register has been updated during the section of subject code" and such a selective swapping (alternating) of the mapping of the speculative "working" and definitive "official" roles is not taught or suggested by Kelly or Le. Further, the dependent claims 2-8, 10 and 11 are each allowable because they depend from an allowable independent claim.